**Sistema de Academia**

**Plano de Gerenciamento de Configuração**

**Versão <1.0>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 05/02/2014 | 1.0 | Criação do Documento | Amanda da Silva |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Índice Analítico**

1. Introdução

1.1 Finalidade

1.2 Escopo

1.3 Definições, Acrônimos e Abreviações

1.4 Visão Geral

2. Ferramentas e Responsábilidades

2.1 Responsabilidades e respectivas Ferramentas

2.2 Ferramentas e informações adicionais

3. Configuração Programa de Gerenciamento de Configuração

3.1 Identificação dos documentos

3.2 Configuração de ambiente

3.2.1 Processamento e Aprovação de Solicitações de Mudança 5

3.2.2 Comitê de Controle de Mudança (CCB) 5

3.3 Estimativa do Status de Configuração 5

3.3.1 Processo de Armazenamento de Mídia e Liberação do Projeto 5

3.3.2 Relatórios e Auditorias 5

4. Marcos 6

5. Treinamento e Recursos 6

6. Controle de Software de Subcontratados e Fornecedores 6

**Plano de Gerenciamento de Configuração**

# Introdução

O plano de gerenciamento de configuração tem por objetivo garantir a integridade durante o ciclo de vida do projeto e o controle das mudanças realizadas.

## Finalidade

A finalidade deste documento é proporcionar maior segurança referente às possíveis mudanças do produto, documentar o projeto, especificar dados de ambiente, informar funções e papeis de cada responsável para garantir que todos os membros da equipe tenham um padrão a seguir e proporcionar ao projeto qualidade e integridade.

## Escopo

Este documento de gerenciamento de configuração aborda os limites do projeto, onde irá determinar as possíveis mudanças, mostrando aos integrantes da equipe como deverão lidar com as mesmas e abordará dados sobre a configuração do sistema, ambiente de armazenamento dos documentos do projeto e a organização de cada documento do projeto.

## 

## Definições, Acrônimos e Abreviações.

|  |  |
| --- | --- |
| Termo | Significado |
| Kanban | Metodologia ágil que permite o controle do fluxo do projeto e o controle da visão geral do status das tarefas do projeto. Esta metodologia não permite a definição de papeis dentro do projeto. |

## Visão Geral

Este documento contém informções referentes a gerência e configuração dos softwares utlizados para o desenvolvimento do projeto, onde mostrará todas as informações necessárias para a configuração do ambiente e a estrutura do mesmo para o desenvolvimento, apresentando a metodologia escolhida, a configuração, a ferramenta utilizada para o gerenciamento de versão, as máquinas e servidores utilizados, fluxos e testes realizados no desenvolvimento do projeto.

# Ferramentas e Responsábilidades

Este capítulo abordará os cargos com suas responsábilidades, ferramentas utilizadas e o conhecimentos necessários.

* 1. Responsabilidades e respectivas ferramentas

|  |  |  |
| --- | --- | --- |
| Cargo | Função | Ferramenta |
| Desenvolvedor | Desenvolver o código do projeto de acordo com a documentação de requisitos. | Conhecer a IDE Eclipse, ferramenta MySQL Workbench e Frameworks Spring MVC e Hibernate. |
| Analista de requisitos | Coletar os requisitos do cliente e criar a documentação do projeto. | Utilizar ferramenta Star UML. |
| QA | Testar a qualidade e a usuabilidade do software. | Utilizar JUnit para realizar os testes. |

* 1. Ferramentas e informações adicionais

|  |  |  |
| --- | --- | --- |
| **Ferramenta** | **Função** | **Informações Adicionais** |
| Eclipse | Ambiente de desenvolvimento em Java para desenvolvimento de aplicações Web. | **Versão:** Juno  **Ferramenta:** Gratuita  **Licença:** Common Public License /  Eclipse Public License |
| Apache Tomcat | Servidor da aplicação WEB. | **Versão:** 7.0  **Ferramenta:** Código aberto  **Licença:** Apache License 2.0 |
| GitHub | Controle de versão dos códigos e das issues. | **Ferramenta**: Site gratuito |
| Git Bash | Cliente Git para Windows. | **Ferramenta:** Gratuita |
| Kanbanize | Site para a realização dos porcedimentos adotados pela metodologia definida no projeto. | **Ferramenta**: Site gratuito |
| Star UML | Ferramenta para a criação dos diagramas do projeto. | **Versão:** 5.0.2.1570  **Ferramenta:** Código aberto. |
| Spring MVC | Framework para o desenvolvimento Web em Java. | **Versão:** 4.0.2  **Framework:** Código aberto |
| Hibernate | Framework para persistência de dados baseados em orientação objeto. | **Versão:** 4.3.1  **Framework:** Código aberto |
| JUnit | Framework para o desenvolvimento dos testes. | **Versão:** 4.11  **Framework:** Código aberto |
| MySQL Workbench | Ferramenta para projeção e manipulação do banco de dados. | **Versão:** 5.2  **Ferramenta:** Gratuita  **Licença:** GPL |
| MYSQL | Banco de Dados do projeto. | **Versão:** ?  **Ferramenta:** Gratuita  **Licença:** GPL |

# Configuração

Este capítulo abordará as normas de identificação do projeto, dados para a configuração da estrutura dos ambientes e informações necessárias para a configuração do controlador de versões.

* 1. Identificação dos documentos

Segue os critérios para a identificações dos documentos, códigos e testes do projeto:

* Documento de controle de configuração de mudanças: ACGCM012014
  + AC – ACademia
  + GCM – Gerência Configuração Mudanças (Sigla do tipo do documento)
  + 01- Número do documento (Sequência)
  + 2014- Ano de criação do documento
* Documento de definição: ACDE012014
  + AC – ACademia
  + DE – Definição (Sigla do tipo de documento)
  + 01- Número do documento (Sequência)
  + 2014 - Ano de criação do documento
* Documentos de requisitos: ACREQ012014
  + AC – ACademia
  + REQ – REQuisitos (Sigla do tipo do docuemto)
  + 01- Número do documento (Sequência)
  + 2014 - Ano de criação do documento
* Diagramas: ACCU012014
  + AC – ACademia
  + CU – Caso Uso (Sigla do tipo do diagrama)
  + 01- Número do documento (Sequência)
  + 2014 - Ano da criação do diagrama
* Pacotes de programas: ACAL2014
  + AC – ACademia
  + AL- ALuno (Sigla da funcionalidade do pacote)
  + 2014 - Ano de criação do pacote

* Testes: ACTU2014
  + AC – ACademia
  + TU- Teste Unitário (Sigla do tipo de teste)
  + 2014 - Ano de criação do teste
  1. Configuração de ambientes

Segue abaixo a tabela abaixo com a estrutura dos ambientes, dados dos equipamentos utlizados e a configuração ulitazada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ambiente | Descrição | Qnt | Configuração de Hardware | Configuração de Software |
| **Desenvolvimento** | Ambiente utilizado para desenvolvimento. | 3 | **Processador:** Intel Core i5  **Memória:** 3G  **Hard Disk:** 1T | Eclipse |
| Apache Tomcat |
| Git Bash |
| Spring MVC |
| Hibernate |
| **Testes** | Ambiente utilizado para a realização dos testes. | 1 | **Processador:** Intel Core i5  **Memória:** 3G  **Hard Disk:** 1T | Jenks |
| JUnit |
| **Banco de Dados** | Ambiente utilizado para o Banco de Dados | 1 | **Processador:** Intel Core i5  **Memória:** 3G  **Hard Disk:** 1T | MySQL |
| MySQL Workbench |
| **Controle de Versões** | Ambiente para o controle de versões do projeto | 1 | **Site GitHubHhHhhhhh** | GitHub |

* 1. Diretórios

|  |  |  |
| --- | --- | --- |
| **Diretório** | **SubDiretório** | **Artefatos** |
| **Link** | Diagramas | Diagrama de Classe |
| Diagrama de Sequência |
| Diagrama de Caso de Uso |
| Diagrama xxxx |
| Documentos | Documentos de Visão |
| Documentos de Requisitos |
| Documentos GCM |
| Códigos Academia | Pacotes de Código |
| Pacotes de Testes |

# Configuração

Este capítulo apresentará informações para a criação, controle e o ciclo de vida da tarefa com os detalhes de todos os status.

* 1. Tarefas e responsabilidades

A descrição a seguir informa o procedimento do ciclo de vida das tarefas do projeto.

Tarefa

* Criar
* Assumir
* Fechar

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|
|
|
|

### Baselines do Projeto

*[As baselines funcionam como um padrão oficial no qual os trabalhos subseqüentes são baseados. Somente mudanças autorizadas podem ser efetuadas nas baselines.*

*Descreva em que pontos do ciclo de vida do projeto ou produto as baselines devem ser estabelecidas. As baselines mais comuns devem ser definidas ao final de cada uma das fases de Iniciação, Elaboração, Construção e Transição. Elas também podem ser geradas no final de iterações ocorridas dentro das várias fases ou com freqüência ainda maior.*

*Descreva quem autoriza uma baseline e o que ela contém.]*

## Controle de Configuração e Mudança

### Processamento e Aprovação de Solicitações de Mudança

*[Descreva o processo pelo qual os problemas e as mudanças são submetidos, revisados e dispostos.]*

### Comitê de Controle de Mudança (CCB)

*[Descreva os membros do CCB e os procedimentos para processar solicitações de mudança e aprovações a serem seguidos pelo CCB.]*

## Estimativa do Status de Configuração

### Processo de Armazenamento de Mídia e Liberação do Projeto

*[Descreva as políticas de retenção e os planos de backup, erros irreversíveis e recuperação. Descreva também como a mídia deve ser mantida — on-line, off-line, tipo de mídia e formato.*

*O processo de liberação descreve o conteúdo do release, a quem ele se destina e se há quaisquer problemas conhecidos ou instruções de instalação.]*

### Relatórios e Auditorias

*[Descreva o conteúdo, o formato e a finalidade dos relatórios e auditorias de configuração solicitados.*

*Os relatórios são usados para avaliar a “qualidade do produto” em qualquer fase do ciclo de vida do projeto ou produto. Os relatórios sobre defeitos com base em solicitações de mudança podem fornecer alguns indicadores de qualidade proveitosos e, dessa forma, alertar a administração e os desenvolvedores para determinadas áreas prioritárias do desenvolvimento. Geralmente os defeitos são classificados por prioridade (alta, média e baixa) e podem ser reportados com base nos seguintes aspectos:*

* *Vencimento (Relatórios Baseados em Períodos): Há quanto tempo defeitos de diversos tipos estão pendentes? Qual é o “tempo de retardo’’ entre o momento em que são encontrados defeitos no ciclo de vida e quando eles são corrigidos?*
* *Distribuição (Relatórios Baseados em Contagens): Existem quantos defeitos nas diversas categorias por proprietário, prioridade ou estado de correção?*
* *Tendência (Relatórios Relacionados a Períodos e Contagens): Qual é o número acumulado de defeitos encontrados e corrigidos no decorrer do tempo? Qual é a classificação dos defeitos detectados e corrigidos? Qual é a “lacuna de qualidade” em termos de defeitos pendentes em comparação com defeitos corrigidos? Qual é a média de tempo de correção de um defeito?]*

# Marcos

*[Identifique os marcos internos e de cliente relacionados ao esforço de CM do projeto ou produto. Esta seção inclui detalhes sobre quando o* ***Plano de Gerenciamento de Configuração*** *deve ser atualizado.]*

# Treinamento e Recursos

*[Descreva as ferramentas de software, o pessoal e o treinamento necessários para implementar as atividades de CM especificadas.]*

# Controle de Software de Subcontratados e Fornecedores

*[Descreva de que forma o software desenvolvido fora do ambiente do projeto será incorporado.]*